

Универзитет у Београду
Електротехнички факултет
Катедра за електронику

Интегрисани рачунарски системи

Лабораторијска вежба

Мултиплексирани LED дисплеј и UART

Београд, 2022.

1 Циљ вежбе

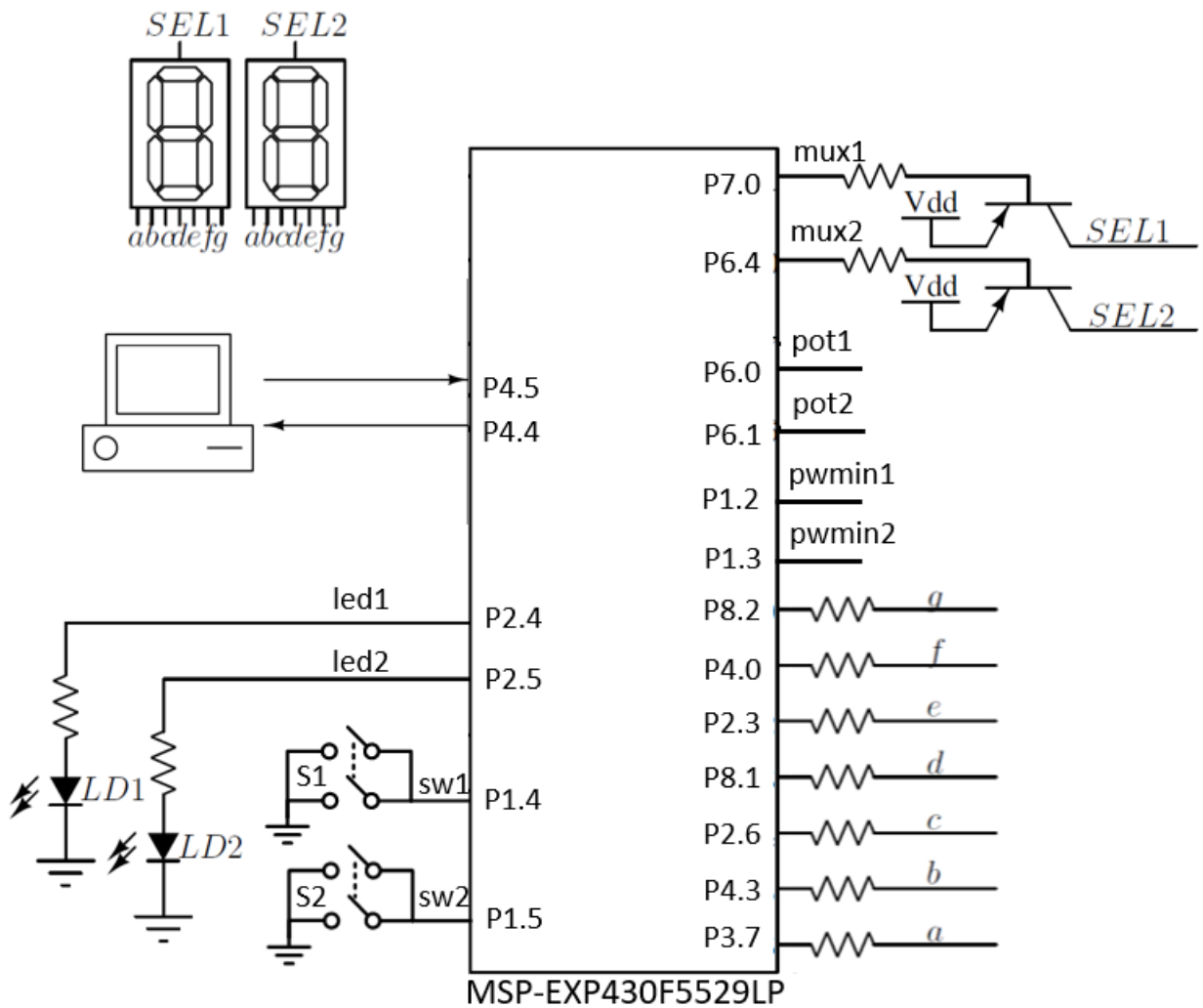
Циљ ове вежбе је да се креирањем неколико С и/или асемблерских програма савладају стандардизовани интерфејс за повезивање мултиплексираног LED дисплеја и да се савлада UART серијска комуникација.

На располагању је софтвер *Code Composer Studio 11*, као и терминал *hterm*, а дизајн се имплементира на плочи са MSP430F5529 микроконтролером.

2 Припрема за рад

Покренути *Code Composer Studio 11* и у жељеном директоријуму креирати *workspace* са називом *lab2*.

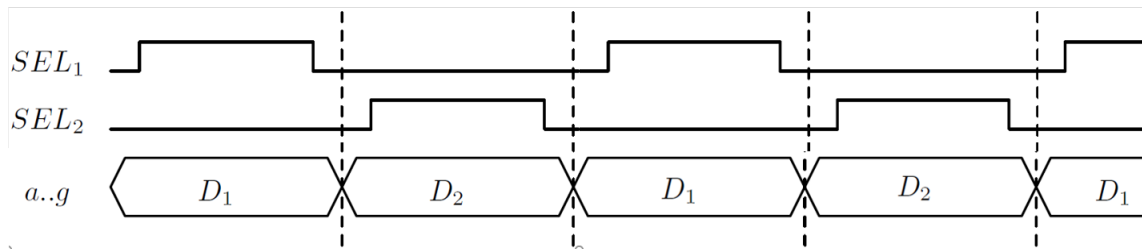
3 Шема развојне плоче са релевантним периферијама



Слика 1: Шема развојне плоче

4 Мултиплексирање дисплеја

На слици 2 приказан је дијаграм сигнала који обезбеђују мултиплексирано исписивање на два LED дисплеја. Сигнали SEL_1 и SEL_2 служе за укључивање сваког од дисплеја понаособ. Да би се на дисплејима исписивао различит садржај потребно је обезбедити да је у једном тренутку укључен само један од њих. Другим речима, потребно је да је само једна од линија $SEL_1 \dots 2$ на нивоу логичке јединице. Садржај који је видљив на једном од дисплеја одговара вредностима постављеним на линије сегмената $a \dots g$ у тренутку док је селекциони сигнал за тај дисплеј активан.



Слика 2: Мултиплексирање дисплеја

5 Задатак

5.1 (10 поена) Написати главни програм у С-у који помоћу прекида тајмера А1 врши мултиплексирање исписа на два седмосегментна LED дисплеја. Садржај дисплеја је одређен вредностима децимално кодираних бинарних цифара дефинисаним у променљивама `disp1` и `disp2`. Потребно је у засебном фајлу креирати табеле константи преко којих се врши пребацивање из BCD у седмосегментни код, као и функцију која на основу BCD вредности активира одговарајуће сегменте. Индекс за табеле је вредност BCD броја из променљивих `disp1` и `disp2`. Тестирати код исписивањем два **различита** броја на дисплеју. Период мултиплексирања одабрати смислено.

У *workspace*-у `lab2` креирати пројекат под називом `lab_glavni`, у којем ћете реализовати тражену функционалност. По завршетку позвати дежурног у циљу провере исправности.

5.2 (15 поена) Релизовати функционалност из тачке **5.1** са прекидном рутином тајмера А1 написаном у асемблеру. Функција која активира одговарајуће сегменте на основу прослеђене BCD вредности се позива из асемблерске прекидне рутине, не треба је реализовати и у асемблерском коду.

У *workspace*-у `lab2` креирати пројекат под називом `lab_asm_isr`, у којем ће се реализовати тражена функционалност. Подешавања С пројекта за рад са асемблерским кодом дата су на вежбама. По завршетку позвати дежурног у циљу провере исправности.

5.3 (25 поена) Модификовати главни програм додавањем иницијализације `USCI_A1` у UART моду, тако да ради на брзини 19200 bps, са 8 бита података, без бита парности и са једним стоп битом (19200,8N1). Потребно је дозволити само пријем преко серијске везе. Пријем реализовати преко С-овског прекида. Примљени карактер се исписује на оба LED дисплеја. Сматрати да се са рачунара шаљу **цифре** у ASCII коду.

По завршетку позвати дежурног у циљу провере исправности. Реализовати тражену функционалност у оквиру пројекта `lab_glavni`.

5.4 (35 поена) Модификовати програм из тачке **5.3**, тако да се имплементира протокол пакетског преноса две цифре. Протокол подразумева пакет следећег облика: `0x73_D1_D2_0x74`. Свака порука почиње стартним бајтом који увек има вредност `0x73` (ASCII код карактера "s"), има два бајта корисничког податка, тј. две цифре и завршава се увек бајтом `0x74` (ASCII код карактера "t").

Порука је успешно примљена ако је карактер "t" примљен као четврти карактер. Освежавање дисплеја вршити само након успешно примљене поруке.¹

Реализовати тражену функционалност у оквиру пројекта `lab_glavni`. По завршетку позвати дежурног у циљу провере исправности.

¹Детаљнија напомена: Решење није исправно ако се чекају четири по четири карактера и онда када стигну четири карактера проверава шта је на првом, а шта на четвртном месту. Уместо тога, потребно је парсирати први карактер и у зависности од тога да ли је он једнак стартном биту омогућити даљи пријем до пуних 4 карактера, или ресетовати бројач. То значи:

- Пријем карактера `s` дозвољава пријем целокупне поруке (још три карактера) и након тога треба проверити вредност четвртог карактера и успешност пријема.
- Уколико се приликом чекања карактера `s` прими неки други карактер, враћа се на чекање карактера `s`.
- Само уколико се након пријема сва четири карактера потврди да је четврти карактер заиста карактер `t`, онда је порука успешно примљена. Тада је потребно освежити дисплеј новим цифрама и вратити се на чекање карактера `s`.
- Сматрати да су други и трећи карактер искључиво цифре 0..9 у ASCII коду.

5.5 (15 поена) Модификовати програм из тачке **5.4** тако да се након успешно примљене поруке покрене процес слања примљене поруке на рачунар ("echo"). Процес слања реализовати коришћењем прекидне рутине `USCI_A1` која се позива када је предајни бафер за слање слободан. Слање првог од 4 карактера иницирати директним уписом у предајни бафер (из главног програма), а остале карактере слати коришћењем прекидне рутине.

Реализовати тражену функционалност у оквиру пројекта `lab_glavni`. По завршетку позвати дежурног у циљу провере исправности.

6 Правила рада

Workspace `lab2` са пројектима `lab_glavni` и `lab_asm_isr` и припадајућим фајловима запаковати у `lab2_gg_bbb.zipovano` (или са другом екстензијом која није `.zip`) и послати мејлом на адресу `cvele@etf.rs` са насловом мејла **13E044IRS_L2**. Алтернативно, позвати дежурног у како би се пројекат исправно сачувао.